

Abstract of the project Grocery Mall Management system:

The Project Supermarket Management System deals with the automation of supermarket. It includes both sales and purchase of items. The project Supermarket Management System is developed with the objective of making the system reliable, easier, fast, and more informative.

There is a lot of reason for the introduction of this project. In the manual System, there are number of inefficiencies that a salesperson faces. Large records-books have to be maintained where relevant and irrelevant information has to be stored which is very untidy and clumsy process.

But our System reduces paper works. On the other hand, there are many inherent problems that exist in any manual system. Usually, they lack efficiency. Less efficiency has a great impact on the productivity of any human being keeping the data up-to-date.

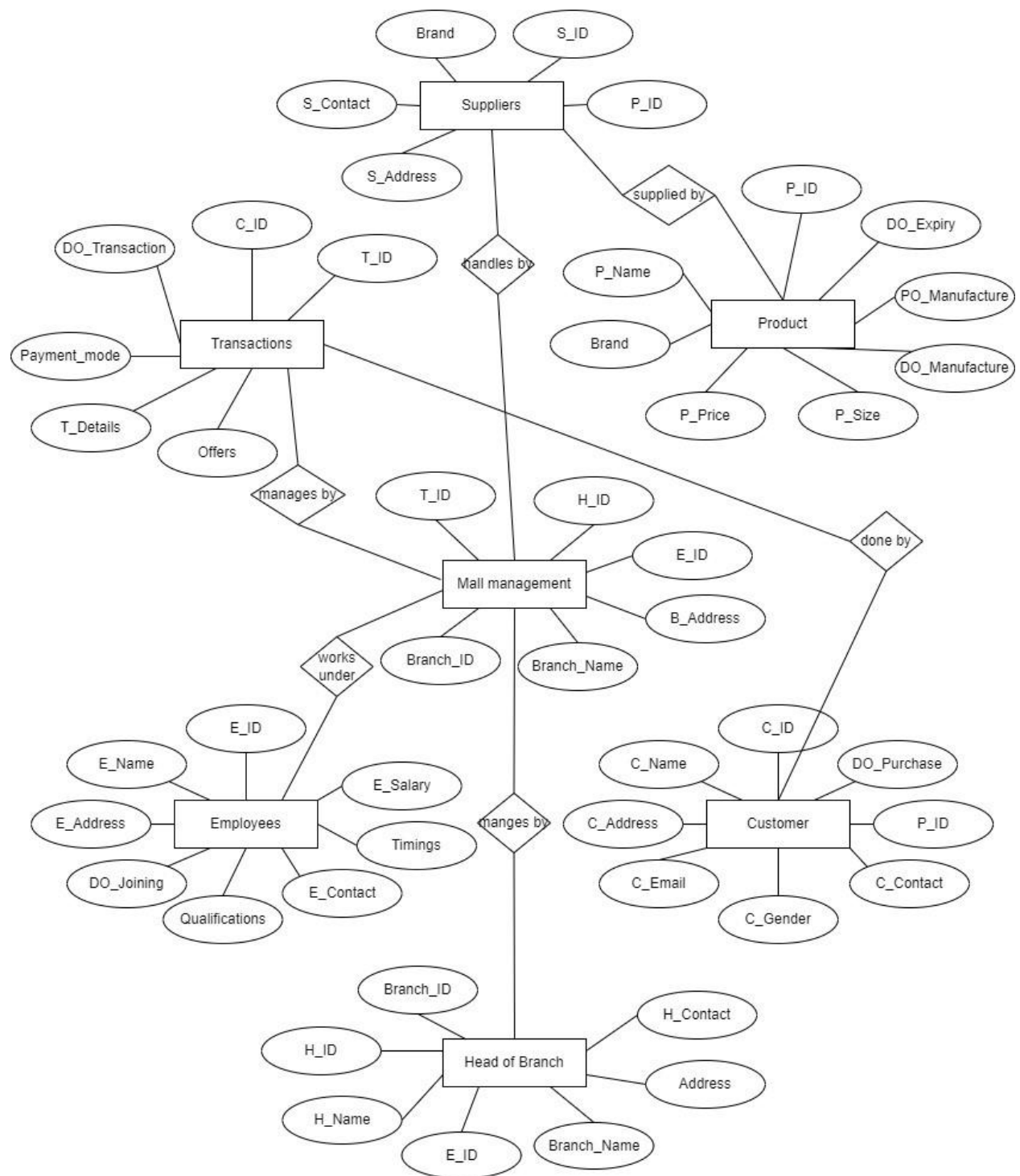
The different modules included in our project are administrative module, employee module, purchase module, sales module and billing module.

Administrator allocates unique username and password to the employees. Each employee can login with the help of his/her unique id and password. Purchase and Sales module contains all the purchase and sales details. All the payment details will be shown in the Billing module.

1.Introduction:-

Detailed design starts after the system design phase is completed. The goal of this phase is to develop the internal logic of each of the modules identified during the system design. In a design document, a more detailed specification is given by explaining in natural language what a module is supposed to do. The design level in which the internal design of the modules, or how the specification of the module can be satisfied, is decided, is often called Detailed design or Logic design. Detailed design essentially expands the system design to contain a more detailed description of the processing logic and data structures so that the design is sufficiently complete for coding. Detailed design is an extension of the system.

ER DIAGRAM:



INFORMATION OF ENTITIES

In total we have seven entities and information on each entity is mentioned below:-

1.Mall_Management :-

(

Attributes:-

Branch_ID, Branch_Name, B_Address

)

The Mall_Management is the management system in which all the information about the particular branch of the company of the Grocery store (Ex:-) will be there. Like Branch_ID (ID of the particular branch), Branch_Name (Name of the branch), B_Address (Address of the particular branch)

2.Head_of_Branch :-

(

Attributes:-

H_ID, H_Name, H_Address, H_Contact

)

The Head_of_Branch is the person who look after the particular branch. Like;

- Who are the employees working in the particular branch
- What are the transactions going on in the branch
- Update about the stock and function the system accordingly

The information(attributes) we will have in our Data base is about his:

ID, Name, Address, Contact, Branch_ID (In which branch he works)

3.Employees :-

(

Attributes:-

E_ID, E_Name, E_Address, E_Contact, E_Salary, E_Timings, DO_Joining, Qualifications

)

Employees are the people who works in the mall. They look after the things where to keep what, they look after the customer who want what, etc. They all are assigned different-different jobs.

The information we should maintain in the Data base with the management is all about them:

ID, name, address, contact, salary he gets every month, timings he who work in a day, date of his joining and his qualifications.

4.Customer :-

(

Attributes:-

C_ID, C_Name, C_Address, C_Contact, C_Email, C_Gender, DO_Purchase, Items_Purchased

)

Customer is the person who purchases any item in the mall. Now a days for many security issues its important to take the information of each and every customer and what are the items he carried with him.

The information we should maintain in the Data base with the management is all about them:

ID, Name, address, contact, email, gender, date of purchased and items purchased.

5.Seller :-

(

Attributes:-

Company_Name, S_Address, S_Contact, S_Offers

)

The is the person from whom be buy the stock for the mall.

The information we should maintain in the Data base with the management is all about them:

Company name, address, contact, offers provided by the particular company.

6.Product :-

(

Attributes:-

*P_ID, P_Name, DO_Manufacture, DO_Expiry, PO_Manufacture, P_Brand,
P_Price, P_Size*

)

We should look after the product brand, date of manufacture, place of manufacture, date of expiry, price of the product and product quantity.

7.Transactions :-

(

Attributes:-

T_ID, DO_Transaction, Payment_mode, T_Details, T_Offers

)

We should look after what are the transactions going on, what are the offers going on at the particular time, date of the particular transaction.

RELATIONSHIP BETWEEN ENTITIES

1. Head_of_Branch and Mall_Management:

Relationship = *"manage by"*

Type of relation = 1 to many

Explanation = In a single company of grocery store there can be many branches and each and every branch has one head. For a company there will be only one main management which look after every branch.

2. Supplier and Mall_Management:

Relationship = "Handle"

Type of relation = Many to Many

Explanation = In a single company Mall_management can many seller at a time, and also a single seller can also sales products to diffrent companies at a time, so it is a many to many relationship.

3. Transaction and Mall_Management:

Relationship = "do"

Type of relation = 1 to Many

Explanation = Many transaction is done my a Mall_management at a time but one Transtion is belong to only one Mall_Management.

3. Transaction and Customer:

Relationship = "do"

Type of relation = 1 to 1

Explanation = That particular transaction can be done by only a single customer and now a days in some malls any one should pay the bill in a single transaction.

4. Employees and Mall_Management:

Relationship = "works under" type

of relation = Many to 1

Explanation = many Employees are working under 1 mall management system. but 1 mall management system belongs to only 1 one employees.

5. Product and Suppliers:

Relationship = "Bought from"

Type of relation = Many to 1

Explanation = That particular product will be brought from a particular company. But that seller sales many products.

6.Product to Customer:

Relationship = "purchased by"

Type of relation = 1 to Many

Explanation = A customer purchases many products, but a particular product can be purchased by a single customer.

RELATIONAL SCHEMAS

Product Table:

- The relation with Product and Seller is many to 1. That's why primary key of Product is used as a foreign key in Seller.
- The relation with Product and Customer is 1 to many. That's why primary key of Product is used as a foreign key in Customer.

Head_of_Branch Table:

- .The relation with Head_of_Branch and Mall_Management is 1 to many. That's why primary key of Head_of_Branch is used as a foreign key in Mall_Management.

Seller Table:

- The relation with Seller and Mall_Management is many to many. That's why primary key of Seller is used as a foreign key in Mall_Management.

Transaction Table:

- .The relation with Transaction and Mall_Management is 1 to many. That's why primary key of Transaction is used as a foreign key in Mall_Management.
- The relation with Transaction and Customer is 1 to 1. That's why the primary key of Transaction is used as a foreign key in Customer.

Employees Table:

- .The relation with Employees and Mall_Management is many to 1. That's why primary key of Employees is used as a foreign key in Mall_Management.

NORMALIZATION

Normalization Rule

Normalization rules are divided into the following normal forms:

1. First Normal Form
2. Second Normal Form
3. Third Normal Form

First Normal Form (1NF)

For a table to be in the First Normal Form, it should follow the following 4 rules:

1. It should only have single (atomic) valued attributes/columns.
2. Values stored in a column should be of the same domain.
3. All the columns in a table should have unique names.
4. And the order in which data is stored, does not matter.

Second Normal Form (2NF)

For a table to be in the Second Normal Form,

1. It should be in the First Normal form.
2. And, it should not have Partial Dependency.

Third Normal Form (3NF)

A table is said to be in the Third Normal Form when,

1. It is in the Second Normal form.
2. And, it doesn't have Transitive Dependency.

NORMALISATION OF GROCERY MALL MANAGEMENT DATABASE:

1.Mall_Management :-

(

Attributes:-

Branch_ID, Branch_Name, B_Address, B_Contact, T_ID, E_ID, H_ID, S_ID, P_ID

)

{Branch_ID} => {Branch_Name} (functional dependency exists, because two different Branch_Name can't have same Branch_Id).

$\{ \text{Branch_ID} \} = > \{ \text{B_Address} \}$ (functional dependency exists).

$\{ \text{Branch_ID} \} = > \{ \text{B_Contact} \}$ (functional dependency exists).

$\{ \text{Branch_ID}, \text{E_ID} \} = > \{ \text{H_ID} \}$ (functional dependency exists).

$\{ \text{Branch_ID}, \text{P_ID} \} = > \{ \text{T_ID} \}$ (functional dependency exists).

- As the attributes of this table does not have sub attributes, it is in first normal form.
- Because every non-primary key attribute is not fully functionally dependent on the primary key of the table this table is not in second normal form.

Hence we have to split the table.

Mall_Management_1 :-

(

Attributes:-

Branch_ID, Branch_Name, B_Address, B_Contact

)

Mall_Management_2 :-

(

Attributes:-

Branch_ID, T_ID, E_ID, H_ID, S_ID, P_ID

)

- Now table is in second normal form.
- Since the table is in second normal form and no non-primary key attribute is transitively dependent on the primary key, the table is now in 3NF.

2.Head_of_Branch :-

(

Attributes:-

H_ID, H_Name, H_Address, H_Contact, Branch_ID, E_ID, E_Salary

)

$\{H_ID\} \Rightarrow \{H_Name\}$ (functional dependency exists, because two different H_Name can't have same H_Id).

$\{H_ID\} \Rightarrow \{H_Address\}$ (functional dependency exists).

$\{H_ID\} \Rightarrow \{H_Contact\}$ (functional dependency exists).

$\{H_ID\} \Rightarrow \{Branch_ID\}$ (functional dependency exists).

$\{H_ID, E_ID\} \Rightarrow \{E_Salary\}$ (functional dependency exists) .

- As the attributes of this table does not have sub attributes, it is in first normal form.
- Because every non-primary key attribute is not fully functionally dependent on the primary key of the table this table is not in second normal form.

Hence we have to split the table.

Head_of_Branch_1 :-

(

Attributes:-

H_ID, H_Name, H_Address, H_Contact, Branch_ID

)

Head_of_Branch_2 :-

(

Attributes:- H_ID,

E_ID

)

- Now table is in second normal form.
- Since the table is in second normal form and no non-primary key attribute is transitively dependent on the primary key, the table is now in 3NF.

3. Employees :-

(

Attributes:-

E_ID, E_Name, E_Address, E_Contact, E_Salary, E_Timings, DO_Joining, Qualifications

)

{E_ID} => {E_Name} (functional dependency exists, because two different H_Name can't have same H_Id).

{E_ID} => {E_Address} (functional dependency exists).

{E_ID} => {E_Contact} (functional dependency exists).

{E_ID} => {E_Salary} (functional dependency exists).

{E_ID} => {E_Timings} (functional dependency exists) .

{E_ID} => {DO_Joining} (functional dependency exists) .

{E_ID} => {Qualifications} (functional dependency exists) .

- As the attributes of this table does not have sub attributes, it is in first normal form.
- Because every non-primary key attribute is fully functionally dependent on the primary key of the table and it is already in first normal form, this table is now in second normal form.
- Since the table is in second normal form and no non-primary key attribute is transitively dependent on the primary key, the table is now in 3NF.

4.Customer :-

(

Attributes:-

C_ID, C_Name, C_Address, C_Contact, C_Email, C_Gender, DO_Purchase, P_ID, T_ID

)

$\{C_ID\} = > \{C_Name\}$ (functional dependency exists, because two different H_Name can't have same H_Id).

$\{C_ID\} = > \{C_Address\}$ (functional dependency exists).

$\{C_ID\} = > \{C_Contact\}$ (functional dependency exists).

$\{C_ID\} = > \{C_Email\}$ (functional dependency exists).

$\{C_ID\} = > \{C_Gender\}$ (functional dependency exists) .

$\{C_ID\} = > \{DO_Purchase\}$ (functional dependency exists) .

$\{C_ID, P_ID\} = > \{T_ID\}$ (functional dependency exists) .

- As the attributes of this table does not have sub attributes, it is in first normal form.
- Because every non-primary key attribute is not fully functionally dependent on the primary key of the table this table is not in second normal form.

Hence we have to split the table.

Customer_1 :-

(

Attributes:-

C_ID, C_Name, C_Address, C_Contact, C_Email, C_Gender, DO_Purchase

)

Customer_2 :-

(

Attributes:-

C_ID, P_ID, T_ID

)

- Now table is in second normal form.
- Since the table is in second normal form and no non-primary key attribute is transitively dependent on the primary key, the table is now in 3NF.

5.Supplier :-

(

Attributes:-

S_ID, Company_Name, S_Address, S_Contact, S_Offers

)

{S_ID} => {S_Name} (functional dependency exists, because two different H_Name can't have same H_Id).

{S_ID} => {S_Company_Name} (functional dependency exists) .

{S_ID} => {S_Address} (functional dependency exists).

{S_ID} => {S_Contact} (functional dependency exists).

{S_ID} => {S_Offers} (functional dependency exists).

- As the attributes of this table does not have sub attributes, it is in first normal form.
- Because every non-primary key attribute is fully functionally dependent on the primary key of the table and it is already in first normal form, this table is now in second normal form.

- Since the table is in second normal form and no non-primary key attribute is transitively dependent on the primary key, the table is now in 3NF.

6.Product :-

(

Attributes:-

P_ID, P_Name, DO_Manufacture, DO_Expiry, PO_Manufacture, P_Brand, P_Price, P_Size

)

{P_ID} => {P_Name} (functional dependency exists, because two different H_Name can't have same H_Id).

{P_ID} => {P_Price} (functional dependency exists).

{P_ID} => {Brand} (functional dependency exists).

{P_ID} => {P_Size} (functional dependency exists).

{P_ID} => {DO_Manufacture} (functional dependency exists) .

{P_ID} => {PO_Manufacture} (functional dependency exists) .

{P_ID} => {DO_Expiry} (functional dependency exists) .

- As the attributes of this table does not have sub attributes, it is in first normal form.
- Because every non-primary key attribute is fully functionally dependent on the primary key of the table and it is already in first normal form, this table is now in second normal form.
- Since the table is in second normal form and no non-primary key attribute is transitively dependent on the primary key, the table is now in 3NF.

7.Transactions :-

(

Attributes:-

T_ID, DO_Transaction, Payment_mode, T_Details, T_Offers, C_ID

)

$\{T_ID\} \Rightarrow \{DO_Transaction\}$ (functional dependency exists, because two different H_Name can't have same H_Id).

$\{T_ID\} \Rightarrow \{Payment_mode\}$ (functional dependency exists).

$\{T_ID\} \Rightarrow \{T_Details\}$ (functional dependency exists).

$\{T_ID\} \Rightarrow \{T_Offers\}$ (functional dependency exists).

$\{T_ID\} \Rightarrow \{C_ID\}$ (functional dependency exists) .

- As the attributes of this table does not have sub attributes, it is in first normal form.
- Because every non-primary key attribute is fully functionally dependent on the primary key of the table and it is already in first normal form, this table is now in second normal form.
- Since the table is in second normal form and no non-primary key attribute is transitively dependent on the primary key, the table is now in 3NF.

TABLES AFTER NORMALIZATION

Mall_Management_1

The screenshot shows the Oracle Live SQL interface. The left sidebar contains navigation options: Home, SQL Worksheet (selected), My Session, Schema, Quick SQL, My Scripts, My Tutorials, and Code Library. The main area is titled "SQL Worksheet" and contains the following SQL code:

```
57 CREATE TABLE Mall_Management_1
58 (
59   "Branch_ID" varchar(20) primary key not null,
60   "Branch_Name" char(30),
61   "B_Address" varchar(40),
62   "B_Contact" number(10)
63 );
64
65 INSERT into Mall_Management_1 VALUES(106,'Ankit','Jalandhar', 9693909676);
66 INSERT into Mall_Management_1 VALUES(107,'jeshaan','Phagwara', 9693059677);
67 INSERT into Mall_Management_1 VALUES(108,'sheela','Jagtial', 9693959078);
68 INSERT into Mall_Management_1 VALUES(109,'mani','Karimnagar', 9690959679);
69 INSERT into Mall_Management_1 VALUES(110,'himanshi','Warangal', 9693919680);
70 select * from Mall_Management_1
71
```

Below the code, a table displays the data inserted into the table:

Branch_ID	Branch_Name	B_Address	B_Contact
106	Ankit	Jalandhar	9693909676
107	jeshaan	Phagwara	9693059677
108	sheela	Jagtial	9693959078
109	mani	Karimnagar	9690959679
110	himanshi	Warangal	9693919680

The bottom of the interface shows the Windows taskbar with the time 10:15 PM on 11/14/2022.

Mall_Management_2

The screenshot shows the Oracle Live SQL interface. The left sidebar contains navigation options: Home, SQL Worksheet (selected), My Session, Schema, Quick SQL, My Scripts, My Tutorials, and Code Library. The main area is titled "SQL Worksheet" and contains the following SQL code:

```
74 CREATE TABLE Mall_Management_2
75 (
76   "Branch_ID" varchar(20) primary key not null,
77   "T_ID" varchar(20) not null references Transaction_,
78   "E_ID" varchar(20) not null references Employee,
79   "H_ID" number(20) not null references Head_of_Branch_1,
80   "S_ID" varchar(20) not null references Supplier,
81   "P_ID" varchar(20) not null references Product
82 );
83 Insert into Mall_Management_2 Values(106,'T001',10253,106545,'S001','P006' );
84 Insert into Mall_Management_2 Values(107,'T002',10252,106523,'S002','P007' );
85 Insert into Mall_Management_2 Values(108,'T003',10258,106562,'S003','P008' );
86 Insert into Mall_Management_2 Values(109,'T004',10251,106591,'S004','P009' );
87 Insert into Mall_Management_2 Values(110,'T005',11259,106510,'S005','P0010');
88 select * from Mall_Management_2
89
```

Below the code, a table displays the data inserted into the table:

Branch_ID	T_ID	E_ID	H_ID	S_ID	P_ID
106	T001	10253	106545	S001	P006
107	T002	10252	106523	S002	P007
108	T003	10258	106562	S003	P008
109	T004	10251	106591	S004	P009
110	T005	11259	106510	S005	P0010

The bottom of the interface shows the Windows taskbar with the time 10:22 PM on 11/14/2022.

Head_of_Branch_1

The screenshot shows the Oracle Live SQL interface. The left sidebar contains navigation links: Home, SQL Worksheet (selected), My Session, Schema, Quick SQL, My Scripts, My Tutorials, and Code Library. The main area is titled "SQL Worksheet" and contains the following SQL code:

```
95  
96 CREATE TABLE Head_of_Branch_1  
97 (  
98   "H_ID" number(20) primary key not null,  
99   "H_Name" varchar(30),  
100  "H_Address" varchar(40),  
101  "H_Contact" number(10),  
102  "Branch_ID" varchar(20) not null references Mall_Management_1  
103 );  
104 INSERT into Head_of_Branch_1 VALUES(106545,'shivank','Khammam', 9693969671,106);  
105 INSERT into Head_of_Branch_1 VALUES(106523,'shwetanshu','Nizamabad', 9693359672,107);  
106 INSERT into Head_of_Branch_1 VALUES(106562,'singh','Amrister', 9693909673,108);  
107 INSERT into Head_of_Branch_1 VALUES(106591,'yusuf','Medchal', 9693950674,109);  
108 INSERT into Head_of_Branch_1 VALUES(106510,'jackson','Manchiryal', 9693959775,110);  
109 select * from Head_of_Branch_1  
110
```

Below the code, a table displays the data inserted into the Head_of_Branch_1 table:

H_ID	H_Name	H_Address	H_Contact	Branch_ID
106545	shivank	Khammam	9693969671	106
106523	shwetanshu	Nizamabad	9693359672	107
106562	singh	Amrister	9693909673	108
106591	yusuf	Medchal	9693950674	109
106510	jackson	Manchiryal	9693959775	110

At the bottom, the footer indicates: © 2022 Oracle - Live SQL 22.3.1, running Oracle Database 19c Enterprise Edition - 19.14.0.0.0 - Database Documentation - Ask Tom - Dev Gym. Built with using Oracle APEX - Privacy - Terms of Use.

Head_of_Branch_2

The screenshot shows the Oracle Live SQL interface. The left sidebar contains navigation links: Home, SQL Worksheet (selected), My Session, Schema, Quick SQL, My Scripts, My Tutorials, and Code Library. The main area is titled "SQL Worksheet" and contains the following SQL code:

```
111  
112  
113 CREATE TABLE Head_of_Branch_2  
114 (  
115   "H_ID" varchar(20) primary key not null,  
116   "E_ID" varchar(20) not null references Employee  
117 );  
118 INSERT into Head_of_Branch_2 VALUES(106545,10253);  
119 INSERT into Head_of_Branch_2 VALUES(106523,10252);  
120 INSERT into Head_of_Branch_2 VALUES(106562,10258);  
121 INSERT into Head_of_Branch_2 VALUES(106591,10251);  
122 INSERT into Head_of_Branch_2 VALUES(106510,11259);  
123 select * from Head_of_Branch_2  
124  
125
```

Below the code, a table displays the data inserted into the Head_of_Branch_2 table:

H_ID	E_ID
106545	10253
106523	10252
106562	10258
106591	10251
106510	11259

At the bottom, the footer indicates: © 2022 Oracle - Live SQL 22.3.1, running Oracle Database 19c Enterprise Edition - 19.14.0.0.0 - Database Documentation - Ask Tom - Dev Gym. Built with using Oracle APEX - Privacy - Terms of Use.

Employee

The screenshot shows the Oracle Live SQL interface. The left sidebar contains navigation options: Home, SQL Worksheet (selected), My Session, Schema, Quick SQL, My Scripts, My Tutorials, and Code Library. The main area is titled 'SQL Worksheet' and contains the following SQL code:

```
38 CREATE TABLE Employee
39 (
40   "E_ID" varchar(20) primary key not null,
41   "E_Name" varchar(30),
42   "E_Address" varchar(40),
43   "E_Contact" number(10),
44   "E_Salary" number(8),
45   "E_Timings" varchar(8),
46   "DO_Joining" varchar(12),
47   "Qualifications" varchar(30)
48 );
49 INSERT into Employee VALUES(10253,'akhil','Jalandhar', 9693959676, 10000, '8 hours','2/12/1990','BTECH');
50 INSERT into Employee VALUES(10252,'jojo','Phagwara', 9693959677, 20000, '10 hours','12/11/1990','BTECH');
51 INSERT into Employee VALUES(10258,'stella','Jagtial', 9693959678, 15000, '8 hours','2/12/1990','BARC');
52 INSERT into Employee VALUES(10251,'monika','Karimnagar', 9693959679, 15000, '8 hours','3/11/1990','BTECH');
53 INSERT into Employee VALUES(11259,'himanshi','Marangal', 9693959680, 10000, '10 hours','2/10/1990','Degree');
54 select * from Employee
```

Below the code, a table displays the data inserted into the 'Employee' table:

E_ID	E_Name	E_Address	E_Contact	E_Salary	E_Timings	DO_Joining	Qualifications
10253	akhil	Jalandhar	9693959676	10000	8 hours	2/12/1990	BTECH
10252	jojo	Phagwara	9693959677	20000	10 hours	12/11/1990	BTECH
10258	stella	Jagtial	9693959678	15000	8 hours	2/12/1990	BARC
10251	monika	Karimnagar	9693959679	15000	8 hours	3/11/1990	BTECH

The bottom of the interface shows the Windows taskbar with the system clock at 10:14 PM on 11/14/2022.

Customer_1

The screenshot shows the Oracle Live SQL interface. The left sidebar contains navigation options: Home, SQL Worksheet (selected), My Session, Schema, Quick SQL, My Scripts, My Tutorials, and Code Library. The main area is titled 'SQL Worksheet' and contains the following SQL code:

```
145 CREATE TABLE Customer_1
146 (
147   "C_ID" number(20) primary key not null,
148   "C_Name" varchar(30),
149   "C_Address" varchar(40),
150   "C_Contact" number(10),
151   "C_Email" varchar(20),
152   "C_Gender" varchar(6),
153   "DO_Purchase" varchar(12)
154 );
155 INSERT into Customer_1 VALUES(150011,'Mark','Hyderabad',9669484525,'jhnn@gmail.com','M','2015-07-19');
156 INSERT into Customer_1 VALUES(150012,'Abdul','Mumbai',9544545735,'gfyj@gmail.com','M','2015-12-24');
157 INSERT into Customer_1 VALUES(150013,'Shivank','Goa',9543656422,'hgjy@gmail.com','M','2015-08-28');
158 INSERT into Customer_1 VALUES(150014,'shweta','Vijayawada',9645531329,'gyhj@gmail.com','M','2015-12-17');
159 INSERT into Customer_1 VALUES(150015,'Shyam','Amaravati',9352631442,'jihu@gmail.com','M','2016-11-22');
160 select * from Customer_1
```

Below the code, a table displays the data inserted into the 'Customer_1' table:

C_ID	C_Name	C_Address	C_Contact	C_Email	C_Gender	DO_Purchase
150011	Mark	Hyderabad	9669484525	jhnn@gmail.com	M	2015-07-19
150012	Abdul	Mumbai	9544545735	gfyj@gmail.com	M	2015-12-24
150013	Shivank	Goa	9543656422	hgjy@gmail.com	M	2015-08-28
150014	shweta	Vijayawada	9645531329	gyhj@gmail.com	M	2015-12-17
150015	Shyam	Amaravati	9352631442	jihu@gmail.com	M	2016-11-22

The bottom of the interface shows the Windows taskbar with the system clock at 10:19 PM on 11/14/2022.

Customer_2

The screenshot shows the Oracle Live SQL interface. The left sidebar contains navigation options: Home, SQL Worksheet (selected), My Session, Schema, Quick SQL, My Scripts, My Tutorials, and Code Library. The main area is titled "SQL Worksheet" and contains the following SQL code:

```
161
162
163 CREATE TABLE Customer_2
164 (
165   "C_ID" number(20) primary key not null,
166   "P_ID" varchar(20) not null references Product,
167   "T_ID" varchar(20) not null references Transaction_
168 );
169 Insert into Customer_2 Values(150011,'P006','T001');
170 Insert into Customer_2 Values(150012,'P007','T002');
171 Insert into Customer_2 Values(150013,'P008','T003');
172 Insert into Customer_2 Values(150014,'P009','T004');
173 Insert into Customer_2 Values(150015,'P0010','T005');
174 select * from Customer_2
175
```

Below the code editor, a table preview shows the data inserted into the Customer_2 table:

C_ID	P_ID	T_ID
150011	P006	T001
150012	P007	T002
150013	P008	T003
150014	P009	T004
150015	P0010	T005

The bottom of the interface shows the Oracle Live SQL version (22.3.1) and the Oracle Database version (19c Enterprise Edition - 19.14.0.0.0).

Supplier

The screenshot shows the Oracle Live SQL interface. The left sidebar contains navigation options: Home, SQL Worksheet (selected), My Session, Schema, Quick SQL, My Scripts, My Tutorials, and Code Library. The main area is titled "SQL Worksheet" and contains the following SQL code:

```
20
21 CREATE TABLE Supplier
22 (
23   "S_ID" varchar(20) primary key not null,
24   "S_Name" varchar(30),
25   "S_Company_Name" varchar(40),
26   "S_Address" varchar(40),
27   "S_Contact" number(10),
28   "S_Offers" varchar(30)
29 );
30 Insert into Supplier Values('S001','Amit','Amazon','sultanpur',9359399492,'20%');
31 Insert into Supplier Values('S002','fatima','google','lucknow',9353452344,'35%');
32 Insert into Supplier Values('S003','sheela','gulgul','shahganj',9359232992,'20%');
33 Insert into Supplier Values('S004','kamal','fipkart','dhobhi',9359392392,'80%');
34 Insert into Supplier Values('S005','Arjun','nozama','khetasarai',9393344392,'50%');
35 select * from Supplier
```

Below the code editor, a table preview shows the data inserted into the Supplier table:

S_ID	S_Name	S_Company_Name	S_Address	S_Contact	S_Offers
S001	Amit	Amazon	sultanpur	9359399492	20%
S002	fatima	google	lucknow	9353452344	35%
S003	sheela	gulgul	shahganj	9359232992	20%
S004	kamal	fipkart	dhobhi	9359392392	80%
S005	Arjun	nozama	khetasarai	9393344392	50%

The bottom of the interface shows the Oracle Live SQL version (22.3.1) and the Oracle Database version (19c Enterprise Edition - 19.14.0.0.0).

Product

The screenshot shows the Oracle Live SQL interface. The left sidebar contains navigation links: Home, SQL Worksheet, My Session, Schema, Quick SQL, My Scripts, My Tutorials, and Code Library. The main area is titled 'SQL Worksheet' and contains the following SQL code:

```
1 CREATE table Product
2 (
3   "P_ID" varchar(20) primary key not null,
4   "P_Name" varchar(30),
5   "DO_Manufacture" varchar(12),
6   "DO_Expiry" varchar(12),
7   "PO_Manufacture" varchar(30),
8   "P_Brand" varchar(20),
9   "P_Price" number(8),
10  "P_Size" varchar(10)
11 );
12 Insert into Product Values('P006','sam','23/12/2002','23/12/2012','para','Amazon',1050,'1 kg');
13 Insert into Product Values('P007','varun','25/12/2003','25/12/2013','bara','gulgul',1100,'1.5 kg');
14 Insert into Product Values('P008','wasim','9/12/2004','9/12/2014','azamgahr','varsi',100,'2 kg');
15 Insert into Product Values('P009','rish','7/12/2005','7/12/2015','jaunpur','benze',105,'5 kg');
16 Insert into Product Values('P0010','Khan','4/12/2007','4/12/2017','khetasarai','ooooops',200,'2 kg');
17 select * from Product
```

Below the code, a table displays the data inserted into the 'Product' table:

P_ID	P_Name	DO_Manufacture	DO_Expiry	PO_Manufacture	P_Brand	P_Price	P_Size
P006	sam	23/12/2002	23/12/2012	para	Amazon	1050	1 kg
P007	varun	25/12/2003	25/12/2013	bara	gulgul	1100	1.5 kg
P008	wasim	9/12/2004	9/12/2014	azamgahr	varsi	100	2 kg
P009	rish	7/12/2005	7/12/2015	jaunpur	benze	105	5 kg
P0010	Khan	4/12/2007	4/12/2017	khetasarai	ooooops	200	2 kg

At the bottom, there is a copyright notice: © 2022 Oracle - Live SQL 22.3.1, running Oracle Database 19c Enterprise Edition - 19.14.0.0.0 - Database Documentation - Ask Tom - Dev Gym. It also mentions 'Built with using Oracle APEX - Privacy - Terms of Use'.

Transaction

The screenshot shows the Oracle Live SQL interface. The left sidebar contains navigation links: Home, SQL Worksheet, My Session, Schema, Quick SQL, My Scripts, My Tutorials, and Code Library. The main area is titled 'SQL Worksheet' and contains the following SQL code:

```
126 CREATE TABLE Transaction_
127 (
128   "T_ID" varchar(20) primary key not null,
129   "DO_Transaction" varchar(12),
130   "Payment_mode" varchar(20),
131   "T_Details" varchar(40),
132   "T_Offers" varchar(30),
133   "C_ID" number(20) not null references Customer_1
134 );
135 Insert into Transaction_ Values('T001','23/12/2012','Amazon pay','1050 rupees','20%',150011);
136 Insert into Transaction_ Values('T002','25/12/2013','google pay','900 rupees','35%',150012);
137 Insert into Transaction_ Values('T003','9/12/2014','gulgul pay','500 rupees','20%',150013);
138 Insert into Transaction_ Values('T004','7/12/2015','fipkart pay','1500 rupees','80%',150014);
139 Insert into Transaction_ Values('T005','4/12/2017','nozama pay','1000 rupees','50%',150015);
140 select * from Transaction_
141
```

Below the code, a table displays the data inserted into the 'Transaction' table:

T_ID	DO_Transaction	Payment_mode	T_Details	T_Offers	C_ID
T001	23/12/2012	Amazon pay	1050 rupees	20%	150011
T002	25/12/2013	google pay	900 rupees	35%	150012
T003	9/12/2014	gulgul pay	500 rupees	20%	150013
T004	7/12/2015	fipkart pay	1500 rupees	80%	150014
T005	4/12/2017	nozama pay	1000 rupees	50%	150015

At the bottom, there is a copyright notice: © 2022 Oracle - Live SQL 22.3.1, running Oracle Database 19c Enterprise Edition - 19.14.0.0.0 - Database Documentation - Ask Tom - Dev Gym. It also mentions 'Built with using Oracle APEX - Privacy - Terms of Use'.

Coding Part

```
-- CREATE table Product

-- (
-- "P_ID" varchar(20) primary key not null,
-- "P_Name" varchar(30),
-- "DO_Manufacture" varchar(12),
-- "DO_Expiry" varchar(12),
-- "PO_Manufacture" varchar(30),
-- "P_Brand" varchar(20),
-- "P_Price" number(8),
-- "P_Size" varchar(10)
-- );

-- Insert into Product Values('P006' , 'sam' , '23/12/2002' , '23/12/2012' , 'para' , 'Amazon' , 1050 , '1
kg');

-- Insert into Product Values('P007' , 'varun' , '25/12/2003' , '25/12/2013' , 'bara' , 'gulgul' , 1100 , '1.5
kg');

-- Insert into Product Values('P008' , 'wasim' , '9/12/2004' , '9/12/2014' , 'azamgahr' , 'varsi' , 100 , '2
kg');

-- Insert into Product Values('P009' , 'rish' , '7/12/2005' , '7/12/2015' , 'jaunpur' , 'benze' , 105 , '5 kg');

-- Insert into Product Values('P0010' , 'Khan' , '4/12/2007' , '4/12/2017' , 'khetasarai' , 'ooooops' , 200 , '2
kg');


-- CREATE TABLE Supplier
-- (
-- "S_ID" varchar(20) primary key not null,
-- "S_Name" varchar(30),
```

```
-- "S_Company_Name" varchar(40),  
-- "S_Address" varchar(40),  
-- "S_Contact" number(10),  
-- "S_Offers" varchar(30)  
-- );
```

```
-- Insert into Supplier Values('S001','Amit','Amazon','sultanpur',9359399492,'20%');  
-- Insert into Supplier Values('S002','fatima','google','lucknow',9353452344,'35%');  
-- Insert into Supplier Values('S003','sheela','gulgul','shahganj',9359232992,'20%');  
-- Insert into Supplier Values('S004','kamal','fipkart','dhobhi',9359392392,'80%');  
-- Insert into Supplier Values('S005','Arjun','nozama','khetasarai',9393344392,'50%');
```

```
-- CREATE TABLE Employee  
-- (  
-- "E_ID" varchar(20) primary key not null,  
-- "E_Name" varchar(30),  
-- "E_Address" varchar(40),  
-- "E_Contact" number(10),  
-- "E_Salary" number(8),  
-- "E_Timings" varchar(8),  
-- "DO_Joining" varchar(12),  
-- "Qualifications" varchar(30)  
-- );
```



```
--
INSERT into Employee VALUES(10253,'akhil','Jalandhar', 9693959676, 10000, '8
hours','2/12/1990','BTECH');

-- INSERT into Employee VALUES(10252,'jojo','Phagwara', 9693959677, 20000, '10
hours','12/11/1990','BTECH');

-- INSERT into Employee VALUES(10258,'stella','Jagtial', 9693959678, 15000, '8
hours','2/12/1990','BARC');

-- INSERT into Employee VALUES(10251,'monika','Karimnagar', 9693959679, 15000, '8
hours','3/11/1990','BTECH');

-- INSERT into Employee VALUES(11259,'himanshi','Warangal', 9693959680, 10000, '10
hours','2/10/1990','Degree');
```

```
-- CREATE TABLE Mall_Management_1
-- (
-- "Branch_ID" varchar(20) primary key not null,
-- "Branch_Name" char(30),
-- "B_Address" varchar(40),
-- "B_Contact" number(10)
-- );
```

```
-- INSERT into Mall_Management_1 VALUES(106,'Ankit','Jalandhar', 9693909676);
-- INSERT into Mall_Management_1 VALUES(107,'jeshaan','Phagwara', 9693059677);
-- INSERT into Mall_Management_1 VALUES(108,'sheela','Jagtial', 9693959078);
-- INSERT into Mall_Management_1 VALUES(109,'mani','Karimnagar', 9690959679);
-- INSERT into Mall_Management_1 VALUES(110,'himanshi','Warangal', 9693919680);
CREATE TABLE Mall_Management_2
```

```
(
-- "Branch_ID" varchar(20) primary key not null,
```

```

--
--

--
    "T_ID" varchar(20) not null references Transaction_,
-- "E_ID" varchar(20) not null references Employee,
-- "H_ID" number(20) not null references Head_of_Branch_1,
-- "S_ID" varchar(20) not null references Supplier,
-- "P_ID" varchar(20) not null references Product
-- );

-- Insert into Mall_Management_2 Values(106,'T001',10253,106545,'S001','P006' );
-- Insert into Mall_Management_2 Values(107,'T002',10252,106523,'S002','P007' );
-- Insert into Mall_Management_2 Values(108,'T003',10258,106562,'S003','P008' );
-- Insert into Mall_Management_2 Values(109,'T004',10251,106591,'S004','P009' );
-- Insert into Mall_Management_2 Values(110,'T005',11259,106510,'S005','P0010');


-- CREATE TABLE Head_of_Branch_1
-- (
-- "H_ID" number(20) primary key not null,
-- "H_Name" varchar(30),
-- "H_Address" varchar(40),
-- "H_Contact" number(10),

```

```

--
--
--

--
  "Branch_ID" varchar(20) not null references Mall_Management_1
);

INSERT into Head_of_Branch_1 VALUES(106545,'shivank','Khammam', 9693969671,106);
-- INSERT into Head_of_Branch_1 VALUES(106523,'shwetanshu','Nizamabad',
9693359672,107);
-- INSERT into Head_of_Branch_1 VALUES(106562,'singh','Amrister', 9693909673,108);
-- INSERT into Head_of_Branch_1 VALUES(106591,'yusuf','Medchal', 9693950674,109);
-- INSERT into Head_of_Branch_1 VALUES(106510,'jackson','Manchiryal', 9693959775,110);


-- CREATE TABLE Head_of_Branch_2
-- (
-- "H_ID" varchar(20) primary key not null,
-- "E_ID" varchar(20) not null references Employee
-- );

-- INSERT into Head_of_Branch_2 VALUES(106545,10253);
-- INSERT into Head_of_Branch_2 VALUES(106523,10252);
-- INSERT into Head_of_Branch_2 VALUES(106562,10258);
-- INSERT into Head_of_Branch_2 VALUES(106591,10251);
-- INSERT into Head_of_Branch_2 VALUES(106510,11259);

```

```
--
```

```
--
```

```
--
```

```
-- CREATE TABLE Transaction_
```

```

--
--
--

--
(
    "T_ID" varchar(20) primary key not null,
    "DO_Transaction" varchar(12),
-- "Payment_mode" varchar(20),
    "T_Details" varchar(40),
-- "T_Offers" varchar(30),
-- "C_ID" number(20) not null references Customer_1
-- );

-- Insert into Transaction_ Values('T001','23/12/2012','Amazon pay' ,1050
rupees','20%','150011');
-- Insert into Transaction_ Values('T002','25/12/2013','google pay' ,900 rupees' ,35%','150012');
-- Insert into Transaction_ Values('T003','9/12/2014' ,gulgul pay' ,500 rupees' ,20%','150013');
-- Insert into Transaction_ Values('T004','7/12/2015' ,fipkart pay','1500 rupees','80%','150014');
-- Insert into Transaction_ Values('T005','4/12/2017' ,nozama pay' ,1000
rupees','50%','150015');


-- CREATE TABLE Customer_1
-- (
-- "C_ID" number(20) primary key not null,
-- "C_Name" varchar(30),
-- "C_Address" varchar(40),
-- "C_Contact" number(10),

```

```
-- "C_Email" varchar(20),  
-- "C_Gender" varchar(6),  
-- "DO_Purchase" varchar(12)  
);
```

```
INSERT into Customer_1  
VALUES(150011,'Mark','Hyderabad',9669484525,'jhnn@gmail.com','M','2015-07-19');  
-- INSERT into Customer_1  
VALUES(150012,'Abdul','Mumbai',9544545735,'gfyj@gmail.com','M','2015-12-24');  
-- INSERT into Customer_1  
VALUES(150013,'Shivank','Goa',9543656422,'hgjy@gmail.com','M','2015-08-28');  
-- INSERT into Customer_1  
VALUES(150014,'shweta','Vijayawada',9645531329,'gyhj@gmail.com','M','2015-12-17');  
-- INSERT into Customer_1  
VALUES(150015,'Shyam','Amaravati',9352631442,'jihu@gmail.com','M','2016-11-22');
```

```
-- CREATE TABLE Customer_2  
-- (  
-- "C_ID" number(20) primary key not null,  
-- "P_ID" varchar(20) not null references Product,  
-- "T_ID" varchar(20) not null references Transaction_  
-- );
```

```
-- Insert into Customer_2 Values(150011,'P006' , 'T001');  
-- Insert into Customer_2 Values(150012,'P007' , 'T002');  
-- Insert into Customer_2 Values(150013,'P008' , 'T003');  
-- Insert into Customer_2 Values(150014,'P009' , 'T004');
```

--

--

-- Insert into Customer_2 Values(150015,'P0010','T005');

----- pl/sql 1-----

-- declare

-- total number(2):=0;

-- begin

-- select count(*) into total

-- from Customer_1;

-- dbms_output.put_line(total);

-- end;

----- pl/sql 2-----

-- declare

-- total number(2):=0;

-- Procedure counts(y out number) is

-- begin

-- select count(*) into y

-- from Product where "P_Price">1000;

-- end;

-- begin

-- counts(total);

```
-- if total >= 2 then
-- dbms_output.put_line('Acceptable '||total);
-- else
-- dbms_output.put_line('Not Acceptable');
```



```
--
```

```
    end if;
```

```
-- end;
```

```
----- pl/sql 3-----
```

```
-- declare
```

```
-- total_row number(2);
```

```
-- begin
```

```
-- update Product
```

```
-- set "P_Price" ="P_Price"+500;
```

```
-- if sql%notfound then
```

```
--    dbms_output.put_line('Data not available');
```

```
-- elsif sql%found then
```

```
--    total_row:=sql%rowcount;
```

```
--    dbms_output.put_line('Total No.. of row is updated is '||total_row);
```

```
-- end if;
```

```
-- end;
```

CONCLUSION

The company wished to replace their manual method with a new database system in order to bring their order system online. Companies may automate their ordering procedures with the help of online systems and databases, as everyone knows, thus developing a database system for a firm will make ordering suitable, useful, and trouble-free. The redundancies and irregularities in the tables are removed using the normalization procedure. We'll begin database construction once all of the tables have been standardized and are redundancy-free. SQL is used to build tables and attributes in this database, as well as to apply constraints to characteristics such as designating the Primary Key and Foreign Key.